# ハイブリッド手法による巡回セールスマン問題の解法

髙橋良英[†]・吉川克哉[††]・小林史和[††]・木村　翼[†††]

# A Hybrid Method of Genetic Algorithms to Solve the Traveling Salesman Problem

Ryouei TAKAHASHI[†], Katsuya YOSHIKAWA[††], Fumikazu KOBAYASHI[††] and Tsubasa KIMURA[†††]

**ABSTRACT**

A hybrid method called Extended Changing Crossover Operator (ECXO) for efficiently obtaining the optimum solution of the Traveling Salesman Problem through flexibly alternating Ant Colony Optimization (ACO) which simulates process of learning swarm intelligence in ants' feeding behavior and Edge Assembly Crossover (EAX) which has been recently noticed as an available method for efficient selection of optimum solution with preserving diversity of chromosomes at any time, is studied. If EAX can not find the optimum solution in the first stage ECXO makes ACO regenerate new chromosomes to merge with the last best solution searched through EAX in order to maintain diversity of chromosomes. Afterwards it makes EAX reproduce better solutions with the merged chromosomes. This trial is repeatedly executed until EAX can find the best solution. In this paper its validity is experimentally verified by using medium-sized TSPs.

***Key Words:*** *TSP, Hybrid Method, ECXO, GA, ACO, EAX*

**キーワード：**巡回セールスマン問題, ハイブリッド手法，拡張遺伝子交叉オペレータ交代法，遺伝的アルゴリズム, アントコロニー最適化法，枝組み立て交叉

## 1. はじめに

巡回セールスマン問題TSP(Traveling Salesman Problem) [1), 2)]の最適解の近似を効率的に得る新しいハイブリッド手法として遺伝的アルゴリズム（Genetic Algorithm (GAs)）[1), 3), 4)]とアントコロニー最適化法 (Ant Colony Optimization (ACO))[5), 6)]の各遺伝子交叉オペレータを任意の時点で交代可能な拡張遺伝子交叉オペレータ交代法（ECXO：Extended Changing Crossover Operators）[7), 8), 9), 10), 11), 12)]を研究する。ECXOでは，ある遺伝子交叉オペレータが探索した巡回路情報を，他の遺伝子交叉オペレータが任意の時点で引継ぎ更新可能である。これにより，巡回路の多様性を確保しつつ効率的に最適な解の探索を可能とする。特に，巡回路の多様性を確保しつつ効率的に最適な解の探索をする手法として近年着目を浴びている枝組み立て交叉（EAX: Edge Assembly Crossover）[13), 14), 15)]を主制御対象の遺伝子交叉オペレータとするECXOを検討した。EAXは両親の巡回路の部分経路長が最短とな

平成23年1月14日受理

†大学院・工学研究科電子電気・情報工学専攻・教授

††工学研究科電子電気・情報工学専攻博士前期課程・2年

†††工学研究科電子電気・情報工学専攻博士前期課程・1年

っていればその部分経路を結合して大域的に最良な解を生成する能力がACOより優れている。しかし，部分巡回路を生成する際に枝の長さに関する情報を利用しないためEAX単独では局所的に最適な解を生成するのに時間がかかる。 ACOは両親の枝の長さに関する情報やフェロモンとして各枝に残された過去の旅の経路長に関する情報から子供が次に訪れる都市を決定するため，生成した経路は局所的に最適となっている。しかし大域的な枝と枝のつながりに関する情報から子供が次に訪れる都市を決定できないため局所最適解に陥りやすいという特徴がある。このため，本ECXOでは比較的早期の世代では遺伝子交叉オペレータACOにより局所的に最適で多様な解を生成し，これを入力として後期の世代では遺伝子交叉オペレータEAXにより大域的に最適な解を効率的に探索する。これまで我々は論文[6), 14)]で参照されているTSPLIB95[16)]のd198, lin318, pcb442によりECXOの有効性を示した[10), 12)]。そのECXOではEAXの性能改善のためACOを唯１回のみ使用した。本論文ではより複雑な問題を解決するためにACOを再帰的に複数回使用してEAXが最適解を探索するECXOの有効性を中規模TSP（att532）で検証したので報告する[11)]。

## 2. 巡回セールスマン問題

あるセールスマンがn箇所の都市を１回ずつ訪問して出発都市に戻ってくる巡回経路のうち，所要距離（または所要費用や所要時間）が最小となる経路を求める問題を巡回セールスマン問題（TSP）という。但しどの２都市間も直接辺で結ばれており，各辺の長さはわかっている。この時，逆順を同一経路と見なして全経路数Ncは（n－１）!/2通りと計算できる。n!はStirlingの公式により，$\sqrt{2\pi}\exp(-n)\times n^{n+1/2}$と近似されるが，この式は網羅的探索で最適解を探索するのに指数オーダ以上の時間がかかることを示している。このような意味でTSPはnに関する多項式時間で解くことが困難なNPクラスに属する問題であると理解されている[17),18)]。

## 3. 拡張遺伝子交叉オペレータ交代法

### 3.1 拡張遺伝子交叉オペレータの特徴

本実験での ECXO の制御対象となった遺伝子交叉オペレータは EAX， ACO でありその機能を C 言語 [19), 20)]で開発した。GAs における個体や ACO における蟻の旅が TSP の巡回路を表現する。実現した各遺伝子交叉オペレータの特徴を以下に整理する。詳細は論文 [10), 12)]を参照されたい。

**(1) 枝組み立て交叉（EAX: Edge Assembly Crossover）**

EAX は親 A と親 B の同数の枝を交換して子供を生成する。親 A の枝を順方向に親 B の枝を逆方向に交互に辿って「A-B サイクル」と呼ばれる幾つかの巡回路を構成する。A-B サイクルを構成する枝を相互に交換して局所的に最適な「緩和個体」と呼ばれる部分巡回路群を親 A と親 B から構成する。最後に２つの部分巡回路間の距離が最短になるように部分巡回路を結合して新たな部分巡回路を生成する。

**(2) アントコロニー最適化手法（ACO: Ant Colony Optimization）**

餌採集時の蟻の集団行動の知能をフェロモンで説明する。過去に旅をした蟻の経路と経路長に関する学習情報を，巡回路上の枝上にフェロモンとして残す。フェロモン量は，蟻が旅行で学習した「巡回路長」の逆数相当の情報であり，後続する蟻は枝上のフェロモン量と枝の短さを考慮して次に訪れる都市を確率的に選択する。本検討での ACO は，局所最適解に陥らないように，エリート蟻のフェロモン量をこれまでの経過時間を考慮して増減させるシミュレーテッドアニ―リング機能を

具備している[9), 17)]。

### 3.2 拡張遺伝子交叉オペレータ交代法 ECXO (ACO→EAX)

#### (1) ねらい

遺伝的アルゴリズムにより TSP の解を得る場合，遺伝子交叉を繰り返すと多様性を失い最良解への収束速度が早くなる現象は GAs における「積み木仮説」の教えるところである[3)]。TSP データが複雑な場合にはこの最良解は最適解ではなく次善解であることが多い。従って，「多様性を保ちながら，最適解への収束速度を向上させる」という，相反する条件を満たしながら最適解を探索する手法，すなわち「積み木仮説」を階層的に適用して最適解を探索する手法，を研究することが TSP を解く時の課題である[13), 14)]。ECXO は，探索経路長とそのばらつきの程度から ACO が生成する遺伝子群の多様性と収束性を定量的に測定監視し EAX への遺伝子交叉オペレータ間の交代時期を自動的に制御することにより課題を解決する。そこでは EAX や ACO や 4 -opt で鎖でつながった Lin-Kernighan 法 CLK[15), 21), 22), 23), 24)]などの機能の異なる複数のオペレータ間で生成遺伝子群を共有することにより多様性を確保しつつ，最適解への収束速度を向上させる方法である。EAX を主制御オペレータする ECXO は以下の考え方でこれを実現する。

（a）ACO が比較的初期の段階で生成した遺伝子群を EAX が引き継ぐことにより EAX の収束速度を向上させる，（b）（TYPE Ⅰ）EAX が既に探索した次善解に，ACO が新たに生成した遺伝子群を混在させ遺伝子群の多様性を確保する。この遺伝子群を再び EAX の入力として解の改良を図る。（b）（TYPE Ⅱ）EAX が探索した次善解を，ACO や CLK を利用して突然変異させる。突然変異させた遺伝子群を再び EAX の入力として解の改良を図る。ECXO 法は（b）を EAX が最適解を探索するまで繰り返す。本実験では（b）(TYPE Ⅰ)の適用結果を述べる。

#### (2) 多様性と収束度による遺伝子交叉オペレータ交代制御

ACO が主遺伝子オペレータ EAX に最新の集団サイズ分の旅行に関する遺伝子情報を Chromosomes というファイル経由で引き継ぐ場合，EAX の収束効率に影響を与えるのは引き継ぐ遺伝子情報の多様性(diversity)と収束度(convergence)である。個体が様々な経路を持つことが多様性であり，それは個体の適応度のばらつきの大きさとなって現れる。このため，多様性を Div=「（現世代を構成する個体の最大適応度－現世代を構成する個体の平均適応度）／（現世代を構成する個体の最大適応度）」で測定する。当 Div は TDGA（Thermo Dynamical Genetic Algorithm）[25)]の TSP への応用尺度の簡易尺度である。多様性が大きければ Div は 1 に近づき，多様性が小さければ Div は 0 に近づく。解が収束するとは現世代までに探索した個体の最大適応度が最適適応度に近づくことであり，その収束度を Conv=「（これまでに探索した個体で最大適応度を持つ個体（最良解）の最大適応度）／（最適適応度）」で測定する。最適適応度とは TSPLIB[33)]で管理されている最短経路を持つ個体（最適解）の適応度と定義する。最適適応度に最良解が近づけば Conv は 1 に近づき，遠のけば Conv は 0 に近づく。ここで個体の適応度は「定数／巡回経路長」で測定する。

多様性と収束度が各々固定値 $\alpha_0$ かつ $\beta_0$ 以上の値をとった時に遺伝子操作オペレータを ACO から EAX に自動的に交代する制御方式を研究することとした。$\alpha_0$，$\beta_0$ の値は一般には集団サイズや扱うデータの複雑さに依存するので，ECXO の走行パラメータとして与えることとした。

### 3.3 拡張遺伝子交叉オペレータ交代法 ECXO (ACO→EAX)のアルゴリズム

Fig. 1 に，ECXO(ACO→EAX)の実現方式を示す。Fig. 1 に示すように，2 段階で最適解を探索する。第 1 段では，先ず ACO で EAX の初期設定時間の

短縮を図る。EAX は ACO が生成した解を Chromosomes というファイルで引き継ぐ。ECXO 法は既に EAX 等が探索した適応度の高い個体（次善解 S1）を ACO 等が比較的初期の世代で生成した個体群に混在させて作った遺伝子ファイルを入力として，EAX を再帰的に起動し解 S1 の改善を図る。このようなテストを独立に 15 回以上行う。第２段では，第１段で生成した解 S1 の併合ファイルを入力として EAX が最適解を探索する。

ECXO のアルゴリズムのアウトラインは以下の通りである。

ECXO のアルゴリズム

＜第１段＞

以下の Family run F(i)を i= 1～Max_Repeat1 + 1 + number of optimum trials 分繰り返す。各 Family run F(i)で EAX が最終的に生成した次善解を S(i)とする。

(第１階層繰り返し)

●seed_id=i としてランダムに初期巡回路を集団サイズ分生成する。

●EAX への引継ぎ条件を，多様性=d0，収束度=c0 として，ACO を起動する。(0≦d0, c0≦1)seed_id=i にて ACO による探索的探索を行う。

●EAX で解を探索する。その際，ACO 解を入力ファイルとする。seed_id=i にて EAX による探索的探索を行う。

●EAX が最適解を探索した場合，i=i+1 として第１段第１階層を繰り返す。

●EAX が最適解を探索しない場合は，以下を j=1～Max_Repat2 分繰り返す。乱数定数＝Const については幾つかの数値を設定し，設定した分第２階層の処理を繰り返す。

（第２階層繰り返し）

●seed_id=(i+j+Const) mod(Max_Repeat1+1)にて，ランダムに初期巡回路を生成する。

●EAX への引継ぎ条件は多様性=d1，収束度=c1 として，ACO を起動する。seed_id=(i+j +Const) mod(Max_Repeat1+1)にて ACO による探索的探索を行う。

●EAX で解を探索する。

その際，ACO 解と EAX が最新に探索した最良解を１つ混在させた遺伝子ファイルを入力ファイルとする。seed_id= i にて EAX による探索を行う。

●EAX が最適解を探索した場合，i=i+1 として第１階層の処理を繰り返す。

●EAX が最適解を探索しない場合は j=j+1 として，第２階層の処理を繰り返す。

＜第２段＞

●第１段で EAX が探索した次善解を混合し遺伝子ファイル S を生成する。S を構成する個体（巡回路）は第１段の S(i)である (i= 1, ・・, Max_Repeat1+1）。

●S から S（i）を除いた遺伝子ファイル W(i+1)を作る。但し W(Max_Repeat1+2)=W(1)である。その際，W(i)は Max_Repeat1 個の個体からなる。

●W(i)を入力として Max_Repeat1 分 EAX を走行し第１段で探索した解を改良する。

パラメータの説明

①Max_Repeat1：本手法が有効であることを統計的に検定するために必要な最低の実験回数である。試験は互いに独立であり，各試行 F(i)毎に最適解またはその次善解 $\alpha$(i)が求まる。また，第１段で最適解が探索できず第２段で EAX が最適解を探索する時の最低の集団サイズを Max_Repeat1 とする。本実験では Dorigo の論文に合わせ，Max_Repeat1=15 とした。

②Max_Repeat1＋１：第１段で最適解を探索できず，第２段で最適解を ECXO が探索できることを統計的に保証するためには，互いに独立な Max_Repeat1 分の試験を第 2 段で行う必要がある。EAX の試験に必要な集団サイズが Max_Repeat1 であることから，第一段で最低限 Max_Repeat1+1 個の個体が生成できれば，そのうち Max_Repeat1 分の個体をランダムに選択する方法は ${}_{\text{Max\_Repeat1+1}}C_{\text{Max\_Repeat1}}$＝Max_Repeat1+1 個なので，第 2 段で最適解を探索できることを統計的に保証することができる。従って第１段での試験回数を Max_Repeat1＋１としている。

③number of optimum trial：第１段での最適解探索

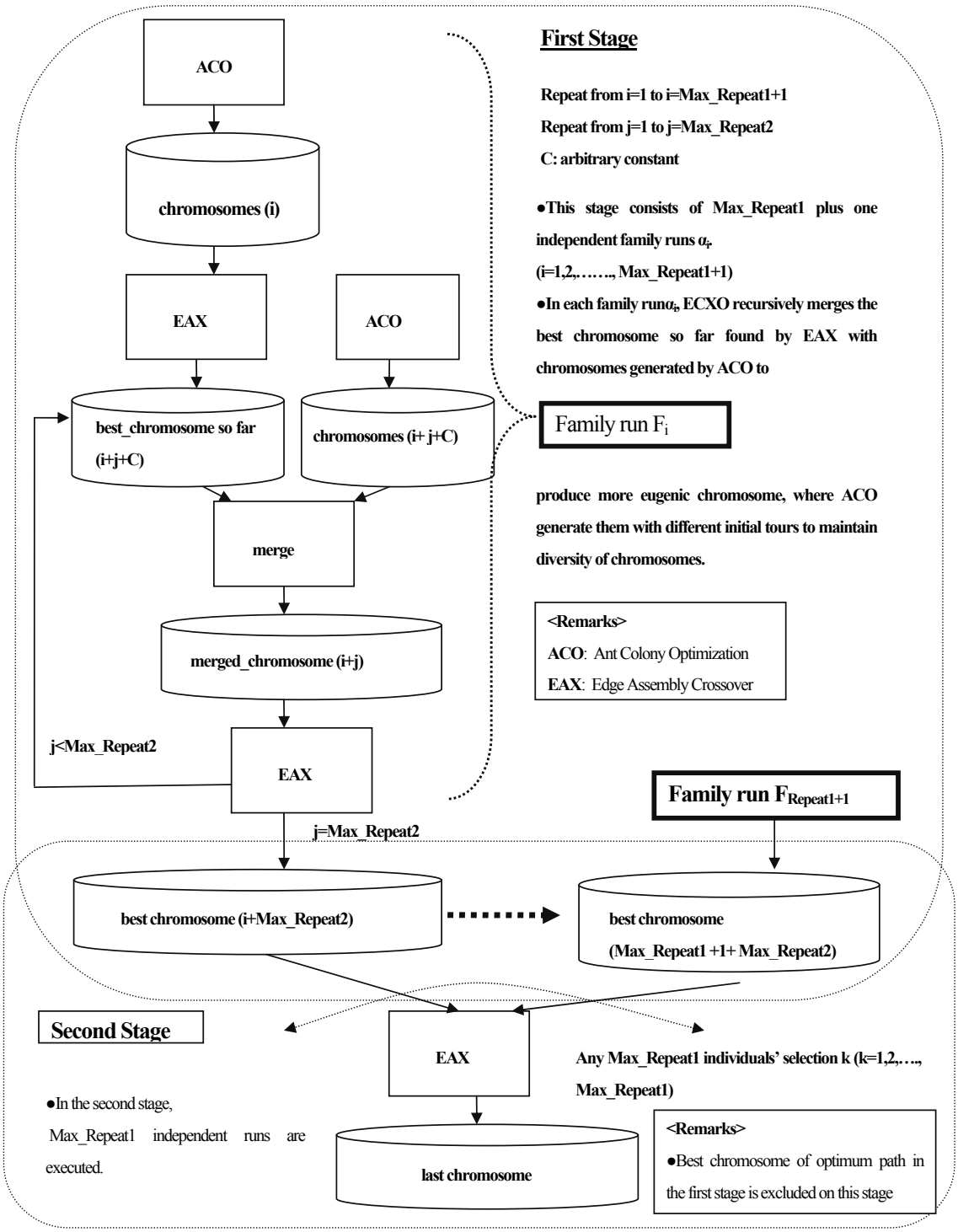


**Fig. 1. ECXO combines EAX with ACO to produce eugenic chromosomes through recursively merging the best chromosome so far found by EAX with chromosomes generated by ACO using different initial tours to maintain the diversity of population.**

回数である。第 1 段階で最適解を探索できなかった分についてのみ第 2 段で最適解の探索を行う必要があるので，結果として②の試行回数＋number of optimum trial 分の試験を行う。

④Max_Repeat2：EAX が局所最適解に陥った場合，この解に ACO により新たに生成した解を混在させたファイルを入力として EAX を再起動し解の改善を図ることができる。その際の ACO と EAX による最大再試行回数が Max_Repeat2 である。

## 4. 実験

### 4.1 実験データ

532 都市問題 att532 を用いて，EAX を主遺伝子交叉オペレータとする EAX の有効性を検証した。TSPLIB によれば，これまでに見つかっている att532 の最短経路長は 27,686 であり，本実験においても ECXO がこれと同じ最短経路を探索した。二都市間の距離は小数点以下四捨五入した整数である [16]。

### 4.2 実験環境

数値実験は，COMPAC NX6320 Genuine Intel(R) CPU, 1.66GHz, 504MB RAM 上の Windows XP 上で行なった。C 言語の処理系は Microsoft Visual C++（R）6.0 デベロプメントシステムであり，Windows XP 上で開発した。

### 4.3 評価対象の遺伝子交叉オペレータ

以下の 3 つの遺伝子交叉オペレータについて機能と性能を評価した。

①EAX，②ACO，③ECXO(ACO→EAX)

### 4.4 プログラム起動パラメータ

(1)ACO の起動パラメータは Dorigo と Stutzle の教科書 [5] によればフェロモン更新タイミングｍは都市数となるが，都市数増大に伴うメモリ負荷を軽減するため，本実験では m=100 とした。その他は教科書にあわせ，フェロモン揮発率 $\rho$ ＝0.3，距離の重み $\beta$ ＝2 とした。

(2)EAX の起動パラメータも同様の理由により，集団サイズ＝100 とした。ACO と EAX の起動パラメータの詳細は論文 [10]を参照されたい。

### 4.5 実験結果

#### (1) ECXO による性能改善効果

EAX，ACO ならびに ECXO モデル（ACO→EAX）の交叉モデルについて，乱数種を 1 から 15 まで変化させて 15 回の独立な試行を行ない，（ｉ）探索した最短経路の中で最も短い経路長 $L_{min}$，（ii）探索した最短経路長の平均値 $L_{average}$，（iii）最適経路探索に成功した回数，（iv）TSBLIB に登録されている最適経路長 $L_{optimal}$ と平均経路長 $L_{average}$ 間の相対誤差 e（=（$L_{average}/L_{optimal}-1$，一般に $L_{min} \geqq L_{optimal}$ ）[16]，各手法において最短経路を探索するのに要した最短探索時間（単位：秒）と平均探索時間（単位：秒）を評価した。ECXO(ACO→EAX)の計算時間は最適解を探索した 13 モデルの第 1 段の計算時間と第 2 段の計算時間の総和についての最小値と平均値である。EAX と ACO の計算時間はそれぞれの最良最短経路長を探索するのに要した計算時間の最小値とその平均値である。

結果を Table 1 にまとめる。

Table1 から以下のことがわかる。

（ｉ）EAX 単独では経路長 27,807 の解しか探索できないこと，ACO 単独では経路長 28,290 の解しか探索できていないこと，

（ii）ECXO（ACO→EAX）が経路長 27,686 の最適経路を探索できていること。

（iii）ECXO の最短経路探索率 0.83=13/15 は高いこと。

（iv）従って，相対誤差 e に関して ECXO は 0.001 と最も小さいこと。EAX は 0.121，ACO が 0.517 であること。

ACO 単独と EAX 単独で最適解が探索できない理由の 1 つは集団サイズやフェロモン更新タ

**Table 1. Comparison of iterative ECXOs with EAX and ACO on best length, average length, the number of optimal trials, relative error, and computer time to find the best length. Results are obtained from fifteen independent trials for att532.**

| No. | Method | Best length | Average length | No. optimum trials | Relative error | Minimum computational time (sec.) | Average computational time (sec.) |
|---|---|---|---|---|---|---|---|
| 1 | EAX | 27,807 | 28,022 | 0 /15 | 0.121 | 4,540 | 5,090 |
| 2 | ACO | 28,290 | 29,116 | 0 /15 | 0.517 | 25,312 | 31,451 |
| 3 | Iterative ECXO(ACO→EAX) | 27,686 | 27,688 | 13/15 | 0.001 | 25,622 | 25,622 |

Relative error is defined as ((Average length)/(Optimal length) – 1)

**Table 2. A Process of iterative ECXO(ACO→EAX) to converge to the optimum solution.**

| cycle | gen. | operators | time (sec) | length |
|---|---|---|---|---|
| 0 | 0 | ACO | 0 | 485,765 |
| 1 | 1 | ACO | 68 | 105,685 |
| | 76 | EAX | 4,883 | 27,800 |
| 2 | 77 | ACO | 4,951 | 98,160 |
| | 99 | EAX | 6,419 | 27,742 |
| 3 | 100 | ACO | 6,487 | 88,724 |
| | 118 | EAX | 7,732 | 27,733 |
| 4 | 120 | ACO | 7,869 | 50,403 |
| | 138 | EAX | 9,103 | 27,721 |
| 5 | 139 | ACO | 9,171 | 89,731 |
| | 149 | EAX | 9,958 | 27,721 |
| 6 | 150 | ACO | 10,026 | 107,115 |
| | 160 | EAX | 10,865 | 27,721 |
| 7 | 162 | ACO | 11,002 | 52,617 |
| | 172 | EAX | 11,799 | 27,721 |
| 8 | 174 | ACO | 11,936 | 52,148 |
| | 195 | EAX | 13,785 | 27,706 |
| 9 | 196 | ACO | 13,853 | 91,501 |
| | 206 | EAX | 14,685 | 27,706 |
| 10 | 207 | ACO | 14,753 | 105,255 |
| | 217 | EAX | 15,573 | 27,706 |
| 11 | 225 | ACO | 16,123 | 39,433 |
| | 235 | EAX | 16,920 | 27,706 |
| 12 | 241 | ACO | 17,333 | 38,915 |
| | 251 | EAX | 18,154 | 27,706 |
| 13 | 259 | ACO | 18,706 | 38,759 |
| | 269 | EAX | 19,542 | 27,706 |
| 14 | 279 | ACO | 20,230 | 39,502 |
| | 289 | EAX | 21,036 | 27,706 |
| 15 | 297 | ACO | 21,586 | 39,276 |
| | 307 | EAX | 22,345 | 27,706 |
| 16 | 319 | ACO | 23,171 | 38,781 |
| | 334 | EAX | 24,235 | 27,703 |
| 17 | 342 | ACO | 24,785 | 39,459 |
| | 352 | EAX | 25,621 | 27,699 |
| 18 | 353 | EAX | 25,622 | 27,686 |

<Remarks> EAX alternates with ACO to find the optimum solution on 18th cycle.

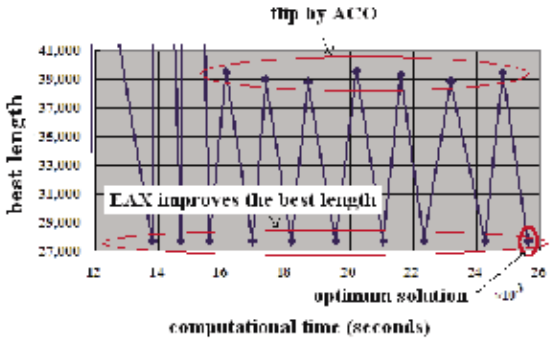


**Fig.2. Process of convergence to the optimal length through iterative ECXO (in a case of att532).**

イミングが小さいことと考えられる。

### (2) ECXO（ACO→EAX）最適解探索プロセス

Table2 は乱数種＝12 で発生させたツアーを初期値として ECXO（ACO→EAX）の最適解探索過程を示している。Table2 は(a)ACO→EAX サイクル番号，(b)世代番号，(c)ACO と EAX のオペレーション種別，(d)コンピュータ時間，(e)その時間までに探索した最短経路長を示している。各世代では 200 個の旅が ACO または EAX により生成される。0 サイクルから 17 サイクルは第 1 ステージ，第 18 サイクルは第 2 ステージである。Table2 から以下のことがわかる。①第 0 世代で最良経路長 485,765 の初期ツアーを生成した後，②第 1 サイクル第 1 世代目完了時 68 秒に ACO は最短経路長 105,685 のツアーを探索し最新の 100 個の旅情報を EAX に引渡していること，③EAX が 76 世代目完了時 4,883 秒に局所最適経路長 27,800 に陥って停止したこと，④第 2 サイクルで新しい初期条件（乱数種=13）で第 77 世代目完了時 4,951

秒に ACO は経路長 98,160 のツアーを生成し最新の 100 個の旅情報を EAX に引渡していること，ACO から引き渡されたこの旅情報とこれまでに EAX が探索した最短経路長 27,800 の旅情報を併合した旅情報を現世代の親として EAX は 99 世代目完了時 6,419 秒時にこれまでより短な経路長 27,742 を探索できたこと，⑤こうして ACO と EAX が交互に遺伝子操作オペレーションを交代し，17 サイクル目 352 世代目 25,621 秒に EAX は局所最短経路長 27,699 を探索していること，⑥このようして探索した他の 14 個の探した最短経路との併合ファイルを入力として EAX は第 2 ステージ 18 世代目完了時 25,622 秒に最適経路長 27,686 を探索した。

Fig.2 は Table2 をグラフ化したものである。グラフの X 軸は実行時間，Y 軸はその時点での探索経路長を示す。グラフは 8 サイクル目 195 世代目以降 EAX と ACO の 2 つの遺伝子交叉オペレータが交互に交代しながら最適解への収束性と集団の多様性を確保しながら最適解を探索している様子を示している。グラフは EAX が次善解に陥っ時，ACO が各サイクルで独立した異なるツアー群を生成し，この生成ツアー群と EAX が探索した次善解を混在させて EAX を再走行させることにより更に短い経路長を探索できる様子を示している。

### 4.6 他の TSP 問題による再帰的 ECXO の有効性の検証

u574 に対してそれぞれ乱数種を変えた独立な 15 回の ECXO 方式の試験を行い att532 と同等以上の性能評価結果を得ている。

以下に最適解探索率を整理する。

内訳： ECXO（ACO→EAX）：14/15= 0.93%，EAX: 0/15=0%，ACO: 0/15=0%.

## 5.結論

拡張遺伝子交叉オペレータ交代法 ECXO（Extended Changing Crossover Operator）の有効性を TSPLIB の中規模 TSP データ（att532,u574）を用いた C プログラム実験で検証した。尚，rat783, rat575, pcb1173 についても同様な結果を得ている。rat575 等の複雑な問題に対しては，ACO と EAX による遺伝子交叉オペレータのみでは最適解が探索できない場合がある。このような場合，解の探索空間の多様性を確保するために Voronoi オペレータ [26]を追加すると最適解探索に成功する場合がある。これについては別の機会に報告する。実験の結果，ACO や EAX 単独では最短経路を探索することが困難なこと， ACO と EAX が交互に交代しながら再帰的に最適解を探索する ECXO（ACO→EAX）により最短経路を探索できることを示した。ACO から EAX への遺伝子群の最適な引継ぎ時期を，引き継ぐ遺伝子群の「多様性」と最適解への「収束度」状況を観測して ECXO は制御している。

### （追記）

本論文は以下の国際会議論文の邦訳である。

11) R. Takahashi: “A Hybrid Method of Genetic Algorithms and Ant Colony Optimization to Solve the Traveling Salesman Problem”, in Proceedings of Eighth International Conference on Machine Learning and Applications, IEEE Computer Society, pp. 81- 88 (2009)

## 参考文献

1) D. E. Goldberg: “Genetic Algorithms in Search, Optimization, and Machine Learning”, Addison-Wesley Publishing Company, Inc. (1989)
2) D.S. Johnson, L. A. McGeoch: “The Traveling Salesman Problem: A Case Study in Local Optimization”, E. H. L.Aarts, J. K. Lenstra, eds., Local Search in Combinatorial Optimization, John Wiley and Sons, Ltd., pp. 215–310 (1997)
3) J. H. Holland: “Adaptation in Natural and Artificial Systems”, MIT Press (1992)
4) N. Sannomiya, H. Kita, H. Tamaoki, T. Iwamoto: “Genetic Algorithms and Optimization”, Asakura-Shoten, 1998 (in Japanese). 三宮信夫, 喜多 一, 玉置 久, 岩本貴司: 「遺伝的アルゴリズムと最適化」, pp.37-51, 朝倉書店 (1998)
5) M. Dorigo and T. Stützle: “Ant Colony Optimization”, The MIT Press (2004)
6) M. Dorigo and L. M. Gambardella: “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem”, IEEE Transactions on Evolutional Computation, 1(1), pp. 53 - 66 (1997)
7) R. Takahashi: “Solving the Traveling Salesman Problem through Genetic

Algorithms with Changing Crossover Operators", Proceedings of Fourth International Conference on Machine Learning and Applications, IEEE Computer Society, pp. 319 - 324 (2005)

8) R. Takahashi and K. Degai: "A Performance Improvement of Genetic Algorithms through Changing Crossover Operators to Solve the Traveling Salesman Problem", Proc. of the 8th International Conference on Computer and Information Technology, organized by Islamic University of Technology, pp. 40 - 45 (2005)

9) R. Takahashi: "A Performance Improvement of Solving the Traveling Salesman Problem through Uniting Changing Crossover Operators to Ant Colony Optimization", Advance in Natural Computation and Data Mining, Proc. of the 2nd International Conference on Natural Computation and the 3rd International Conference on Fuzzy Systems and Knowledge Discovery, Xidian University, pp. 114 - 130 (2006)

10) R. Takahashi, "Solving the Traveling Salesman Problem through Extended Changing Crossover Operators", The Institute of Electrical Engineers of Japan, IEEJ Trans. EIS, Vol. 128, No. 12, pp.1820-1832, 2008 (in Japanese), 高橋良英:「拡張遺伝子交叉オペレータ交代法による巡回セールスマン問題の解法」, 電気学会,Vol.128,No.12,Sec. C., pp.1820-1832 (2008)

11) R. Takahashi: "A Hybrid Method of Genetic Algorithms and Ant Colony Optimization to Solve the Traveling Salesman Problem", in Proceedings of Eighth International Conference on Machine Learning and Applications, IEEE Computer Society, pp. 81- 88 (2009)

12) R. Takahashi: "Extended Changing Crossover Operators to Solve the Traveling Salesman Problem", Electronics and Communications in Japan, Vol.93, NO. 7, pp. 1- 16 (2010)

13) Y. Nagata and S. Kobayashi: "The proposal and evaluation of a crossover for Traveling Salesman Problem: Edge Assembly Crossover", JSAI, Vol.14,No.5,pp.848-859, 1999 (in Japanese). 永田裕一,小林重信:「巡回セールスマン問題に対する交叉：枝組み立て交叉の提案と評価」, 人工知能学会誌,Vol.14,No.5,pp.848-859 (1999)

14) M.-K.Tsai, J.-M.Yang, Y.-F.Tsai, C.-Y. Kao: "Some issues of designing genetic algorithms for traveling salesman problem", Soft Computing, No.8, pp.689-697 (2004).

15) M.-K.Tsai, J.-M.Yang, Y.-F.Tsai, C.-Y. Kao: "An Evolutionary Algorithm for Large Traveling Salesman Problems", IEEE Transactions on Systems, Man, and Cybernetics- Part B. Cybernetics., Vol. 34, No.4, pp. 1718 – 1729 (2004)

16) TSPLIB95: http://www.informatik.uni-heidelberg.de/

17) E. Aarts and J. K. Lenstra: "Local Search in Combinatorial Optimization", Princeton University Press (2003)

18) M. Garey and D. Johnson: "Computers and Intractability, A Guide to the Theory of NP-Completeness", W. H Freeman and Company (1979)

19) B. W. Kernighan and D.M. Richie: "The C Programming Language, Second Edition", Bell Telephone Laboratories (1988)

20) H. Hirano: "A Genetic Algorithm C programming", Personal Media Inc., pp. 232-238, 1995 (in Japanese). 平野広美:「遺伝的アルゴリズム C プログラミング」, pp.232-pp.238, パーソナルメディア社 (1995).

21) K. Helsgaun: "An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristics", European Journal of Operational Research, 126(1), pp. 106 - 130 (2000)

22) S. Lin & B. W. Kernighan: "An Effective Heuristic Algorithm for the Traveling-Salesman Problem", Oper. Res. 21,pp.498-516 (1973)

23) D. Applegate, R. Bixby, V. Chvatal, W. Cook: "Finding Tours in the TSP", Tech. Rep. TR99-05, Dept. Computat. Appl. Math., Rice Univ., Houston, TX 77005 (1999)

24) O. Martin, S. W. Otto, E. W. Felten: "Large-Step Markov Chains for the TSP Incorporating Local Search Heuristics", Operation Research Letters, vol. 11, pp.219-224 (1992).

25) 前川,玉置,喜多,西川：熱力学的選択ルールを用いた巡回セールスマン問題の遺伝的解法, 計測自動制御学会誌論文集, Vol. 33, No. 9, pp. 939-946 (1997).

26) Dong-II Seo and Byung-Ro Moon, Voronoi: "Quantized Crossover for Traveling Salesman Problem, Genetic and Evolutionary, Computation Conference, pp.544-552 ( 2002)